



PATENT ABSTRACTS OF JAPAN

(11) Publication number: 09069047 A

(43) Date of publication of application: 11.03.97

(51) Int. Cl

G06F 9/38**G06F 9/38****G06F 9/46****G06F 15/78**

(21) Application number: 07225036

(71) Applicant: SONY CORP

(22) Date of filing: 01.09.95

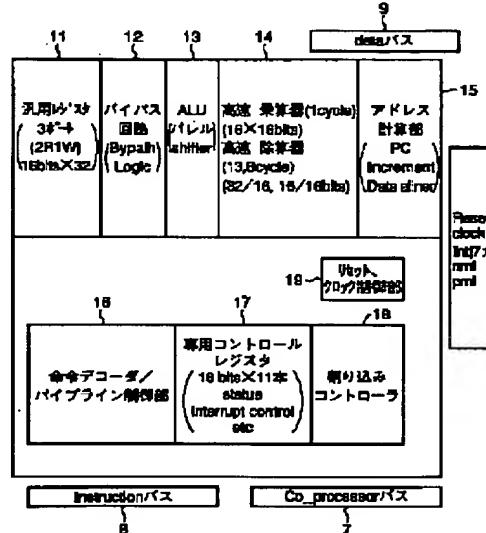
(72) Inventor: GOTO MASARU

(54) RISC TYPE MICROPROCESSOR AND INFORMATION PROCESSOR

(57) Abstract:

PROBLEM TO BE SOLVED: To enable the RISC type microprocessor to perform a fast interruption processing and a numerical arithmetic processing.

SOLUTION: A general register 11 stores an instruction of 16-bit fixed length and a by-pass circuit 12 provides the result of a comparison instruction speedily at the time of next conditional branch instruction execution. An ALU 13 performs a logical arithmetic process, etc., and a fast multiplier and fast divider 14 performs the fast numerical arithmetic processing. An address calculation part 15 performs address calculation, an instruction decoder and pipeline control part 16 decodes instructions and perform pipeline control. A dedicated control register 17 is used as an interrupt stack pointer, etc., and an interruption controller 18 performs multiple interruption processes. A coprocessor bus 7 is provided independently of a data bus 9.



COPYRIGHT: (C)1997,JPO

(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開平9-69047

(43)公開日 平成9年(1997)3月11日

(51) Int.Cl. ⁶	識別記号	府内整理番号	F I	技術表示箇所
G 06 F	9/38	310	G 06 F	9/38
		370		310 A
	9/46	311		370 C
	15/78	510	9/46	311 A 06, C7
			15/78	510 G

審査請求 未請求 請求項の数9 OL (全11頁)

(21)出願番号 特願平7-225036

(22)出願日 平成7年(1995)9月1日

(71)出願人 000002185

ソニー株式会社

東京都品川区北品川6丁目7番35号

(72)発明者 後藤 勝

東京都品川区北品川6丁目7番35号 ソニ
ー株式会社内

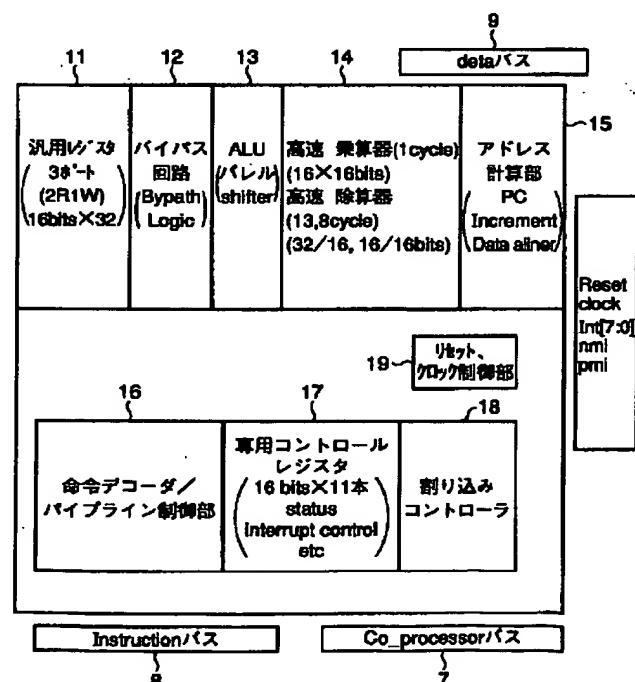
(74)代理人 弁理士 稲本 義雄

(54)【発明の名称】 RISC型マイクロプロセッサおよび情報処理装置

(57)【要約】

【課題】 RISC型マイクロプロセッサにおいて、高速な割り込み処理と数値演算処理を行うことができるようとする。

【解決手段】 汎用レジスタ11は16ビット固定長の命令を記憶し、バイパス回路12は比較命令の結果を迅速に次の条件分岐命令実行時に提供する。ALU13は論理演算処理等を行い、高速乗算器／高速除算器14は高速に数値演算を行う。アドレス計算部15はアドレス計算を行い、命令デコーダ／パイプライン制御部16は、命令のデコードとパイプライン制御を行う。専用コントロールレジスタ17はインタラプトスタックポインタなどとして使用され、割り込みコントローラ18によって多重の割り込み処理が実行される。コプロセッサバス7は、データバス9とは独立して設けられている。



【特許請求の範囲】

【請求項1】 パイプライン制御を行うRISC型マイクロプロセッサにおいて、所定の固定長の命令をデコードするデコード手段を備えることを特徴とするRISC型マイクロプロセッサ。

【請求項2】 前記命令の長さは16ビットであることを特徴とする請求項1に記載のRISC型マイクロプロセッサ。

【請求項3】 第1の命令の実行結果に対応する情報を前記第1の命令の次に実行される第2の命令に提供する情報提供手段をさらに備えることを特徴とする請求項1に記載のRISC型マイクロプロセッサ。

【請求項4】 前記命令には、ビット処理を行う命令が含まれることを特徴とする請求項1に記載のRISC型マイクロプロセッサ。

【請求項5】 前記命令には、即値演算を行う命令が含まれることを特徴とする請求項1に記載のRISC型マイクロプロセッサ。

【請求項6】 パイプライン制御を行うRISC型マイクロプロセッサにおいて、複数の割り込みがあったとき、前記割り込みに対応する処理をそれぞれ所定の時間ずつ巡回的に実行する第1のモードと、前記割り込みに対応する処理を逐次実行する第2のモードのいずれかのモードを指定するモード指定手段と、

前記モード指定手段により指定された前記モードに従って、割り込み処理を制御する割り込み制御手段とを備えることを特徴とするRISC型マイクロプロセッサ。

【請求項7】 前記割り込み制御手段は、前記割り込みがあったとき、必要情報をスタックに退避し、割り込み処理が終了したとき、前記必要情報をスタックから取り出すことを特徴とする請求項6に記載のRISC型マイクロプロセッサ。

【請求項8】 コプロセッサが接続されたRISC型マイクロプロセッサにおいて、前記RISC型マイクロプロセッサのデータバスとは独立して、前記コプロセッサが専用に使用し、データの伝送を行う伝送手段を備えることを特徴とするRISC型マイクロプロセッサ。

【請求項9】 GPS衛星からの電波を受信する受信装置と、前記受信装置により受信された前記電波に対応する信号を処理するRISC型マイクロプロセッサからなる情報処理装置において、

前記GPS衛星からの電波を受信する受信装置と、前記受信装置により受信された前記電波に対応する信号を処理する前記RISC型マイクロプロセッサが1チップ化されていることを特徴とする情報処理装置。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】 本発明は、RISC型マイク

ロプロセッサおよび情報処理装置に関し、特に、従来の1チップマイコンにRISCテクノロジを導入することにより、割り込み処理や数値演算を高速に実行できるようにしたRISC型マイクロプロセッサおよび情報処理装置に関する。

【0002】

【従来の技術】 従来のRISC (Reduced Instruction Set Computer) は、WS (Work Station) などのコンピュータのエンジンとして、演算速度の高速性のみを追求して発展してきた。また、その命令長は通常32ビット固定長である。

【0003】

【発明が解決しようとする課題】 このため、従来のRISCを用いて1チップ化された1チップマイコン (マイクロコンピュータ) においては、コード効率が悪く、割り込み処理は別チップにおいて処理されるため、通常、割り込み処理のスピードが遅い課題があった。一方、従来の1チップマイコンにおいては、高速演算が不可能であった。

【0004】 本発明はこのような状況に鑑みてなされたものであり、1チップマイコンにRISCテクノロジを取り入れ、高速な割り込み処理を可能にするとともに、高速な数値演算をも可能にするものである。

【0005】

【課題を解決するための手段】 請求項1に記載のRISC型マイクロプロセッサは、所定の固定長の命令をデコードするデコード手段を備えることを特徴とする。

【0006】 請求項6に記載のRISC型マイクロプロセッサは、複数の割り込みがあったとき、割り込みに対応する処理をそれぞれ所定の時間ずつ巡回的に実行する第1のモードと、割り込みに対応する処理を逐次実行する第2のモードのいずれかのモードを指定するモード指定手段と、モード指定手段により指定されたモードに従って、割り込み処理を制御する割り込み制御手段とを備えることを特徴とする。

【0007】 請求項8に記載のRISC型マイクロプロセッサは、RISC型マイクロプロセッサのデータバスとは独立して、コプロセッサが専用に使用し、データの伝送を行う伝送手段を備えることを特徴とする。

【0008】 請求項9に記載の情報処理装置は、GPS衛星からの電波を受信する受信装置と、受信装置により受信された電波に対応する信号を処理するRISC型マイクロプロセッサが1チップ化されていることを特徴とする。

【0009】 請求項1に記載のRISC型マイクロプロセッサにおいては、デコード手段により、所定の固定長の命令がデコードされる。従って、命令のデコード処理を高速化することができる。

【0010】 請求項6に記載のRISC型マイクロプロセッサにおいては、モード指定手段により、複数の割り

込みがあったとき、割り込みに対応する処理をそれぞれ所定の時間ずつ巡回的に実行する第1のモードと、割り込みに対応する処理を逐次実行する第2のモードのいずれかのモードが指定され、このモードに従って、割り込み制御手段により割り込み処理が制御される。従って、効率よく割り込み処理を実行させることができる。

【0011】請求項8に記載のRISC型マイクロプロセッサにおいては、RISC型マイクロプロセッサのデータバスとは独立して、コプロセッサが専用に使用し、データの伝送を行う伝送手段が設けられる。従って、コプロセッサは、RISC型マイクロプロセッサとは独立してデータ伝送を行うことができる。

【0012】請求項9に記載の情報処理装置においては、GPS衛星からの電波を受信する受信装置と、受信装置により受信された電波に対応する信号を処理するRISC型マイクロプロセッサが1チップ化されている。従って、装置を小型化することができる。

【0013】

【発明の実施の形態】図1は、本発明のRISC型マイクロプロセッサを応用した1チップマイコン（マイクロコンピュータ）の一実施例の構成を示している。1チップマイコンを構成するRISC（Reduced Instruction Set Computer）1には、8本の外部割り込み（Interrupt）と、ノンマスカブルの割り込みnmi（Non Maskable Interrupt）およびpmi（Power Management Interrupt）が入力可能にされている。さらに、外部からリセット（Reset）信号とクロック信号（Clock）が入力されるようになされている。

【0014】RISC1は、リセット信号が入力されると、IPL（Initial Program Loader）-ROM3に格納されているプログラムを実行する。このプログラムが実行されると、コプロセッサ（Co_pro）2のレジスタを通して、メモリコントローラ（Memory Controller）5が起動され、コードバッファ（Code buffer）4およびデータバッファ（Data buffer）5、および図示せぬメインメモリに、図示せぬ補助記憶装置からオペレーティングシステムがロードされるようになされている。

【0015】図2は、図1に示したRISC1の内部の詳細な構成例を示している。汎用レジスタ11は、32本の16ビット固定長のレジスタにより構成されている。バイパス回路12（情報提供手段）は、比較命令が実行されたときの結果に対応するフラグ情報を、次の条件分岐命令で即使用可能にするようになされている。

【0016】ALU（Arithmetic and Logic Unit）は、パレルシフタ等で構成され、所定の論理演算や数値演算を行うようになされている。高速乗算器／高速除算器14は、例えば、16ビットのデータと16ビットのデータの乗算処理を1サイクルで実行し、32ビットのデータの16ビットのデータによる除算処理を13サイクルで実行し、16ビットのデータの16ビットのデータに

よる除算処理を8サイクルで実行するようになされている。

【0017】アドレス計算部15は、プログラムカウンタなどにより、アドレス計算を行う。命令デコーダ／パイプライン制御部16（デコード手段）は、メモリから読み出した例えば16ビット固定長の命令を解読し、5段のパイプラインによってパイプライン処理を行うようになされている。

【0018】専用コントロールレジスタ17（モード指定手段）は、11本の16ビット長のレジスタからなり、ステータス情報（フラグ情報）や割り込み制御情報などを記憶する。割り込みコントローラ18（割り込み制御手段）は、専用コントロールレジスタ17に記憶された割り込み制御情報に基づいて、割り込み処理をコントロールするようになされている。リセット／クロック制御部19は、リセット信号とクロック信号を発生し、各部に供給するようになされている。

【0019】図3は、図1に示した1チップマイコンのレジスタマップを示している。この場合、2つのコプロセッサ（COP0, COP1）が拡張されたものとする。RISC（CPU）1は32個の汎用レジスタを有しており、第1コプロセッサ（COP0）2の32個のコントロールレジスタをCPU内部のコントロールレジスタとして使用している。すなわち、マッピングしている。そのため、ユーザは、残りの32個のコントロールレジスタだけを使用することができる。図示せぬ第2コプロセッサ（COP1）の場合、ユーザは何の制約もなく64個のコントロールレジスタを使用することができる。

【0020】この汎用レジスタは、特殊な命令以外は、演算のレジスタとして使用することができる。そのうち、レジスタACC（R1）は、即値、ビット処理系のオペランドとして使用可能である。また、例外として、レジスタSP（R30）は、例えば「RET」命令実行時のスタックポインタとして使用される。さらに、レジスタISP（R31）は、割り込み処理、例外処理、および「RETI」命令実行時のスタックポインタとして使用される。これらの汎用レジスタ（レジスタACC、レジスタSP、およびレジスタISPを含む）の初期値40は不定とされる。

【0021】図3に示したように、第1コプロセッサ（COP0）2のレジスタG31乃至G0およびC31乃至C0、第2コプロセッサ（COP1）のレジスタG31乃至G0およびC31乃至C0の合計128個のコプロセッサレジスタを拡張することができる。コプロセッサレジスタ、汎用レジスタ、およびメモリとの間の転送は命令によって定義されている。

【0022】上記コプロセッサレジスタのうち、第1コプロセッサのレジスタC31乃至C0と、第2コプロセッサのレジスタG31乃至G0およびC31乃至C0の

合計96個のコプロセッサレジスタを外部に拡張することができる。第1コプロセッサのレジスタG31乃至G0は、CPUのコントロールレジスタとして機能するものを11個内蔵している。使用していない21個のレジスタは将来の拡張用である。

【0023】SR (Status Register) は、演算結果などに対応するフラグを保持する。MCR (Machine Control Register) は、マシン制御に使用される。例えば、32ビットの割り算を可能にするか否かを指示するデータが設定される。IBR (Interrupt Base register) には、割り込みベクタテーブルのベースアドレスが設定される。JBR (Jump Base Register) は、特殊なジャンプ命令が実行され、ブランチするとき使用される。ICR (Interrupt Control Register) は、割り込みの制御に用いられ、3重の割り込み許可のスタックを構成している。

【0024】IMR0 (Interrupt Mode Register0) は、外部割り込みモードの制御を行うとき使用される。例えば、後述する「ラウンドロビン」モードや「フィックス」モードが設定される。IMR1 (Interrupt Mode Register1) は、8本の外部割り込みをグループ分けするときなどに使用される。DAB0 (Data Address Break0) には、データバッファのブレークアドレスが設定される。IAB1 (Instruction Address Break1) には、命令バッファのブレークアドレスが設定される。IAB2 (Instruction Address Break2) には、命令バッファのブレークアドレスが設定される。XDDD (Extended Divide Divident) には、拡張除算命令の実行時、32ビットの被除数の上位16ビットがセットされる。

【0025】また、ユーザが拡張可能なオペレーションコードは、第1コプロセッサ、第2コプロセッサそれぞれ30個程あり、アセンブラーの「DW」命令により簡単に実行することができる。また、CPUは、コプロセッサを使用していないときの例外処理をサポートしており、コプロセッサのハードウェアをソフトウェアによってエミュレーションすることが可能である。

【0026】図4は、RISC1のアドレス空間の構成例を示している。このように、インストラクションとデータそれぞれ独立に64キロバイト(KB) 使用することができ、外部コプロセッサとのやりとりには、96ワードの外部レジスタを使用することができる。外部リセットを受け付けると、FF60Hへ分岐し、そこに格納されている命令を実行する。ベクタアドレスは、2ワード(4バイト)おきに設定され、最初の1ワードにブランチ(ジャンプ)命令が設定され、後の1ワードはディレイスロットであり、「nop (ノーオペレーション)」命令または他の命令が設定される。

【0027】ベクタアドレスはレジスタIBRの設定値(上位8ビット)をベースアドレスにしており、256バイト境界で任意の位置に配置することができる。ベク

タアドレスには分岐先のオフセット値が設定され、このオフセット値とレジスタIBRに格納されているベースアドレスに基づいて、例えば、両者の論理和を演算することにより、分岐先のアドレスが決定される。

【0028】図5は、図1に示した1チップマイコンで使用される命令のフォーマットの例を示す図である。図5(a)は、MOV命令のフォーマットを示している。このように、命令長は16ビットの固定長であり、オペレーションコード(OP_CODE)と2つのオペランド(SRC1/DEST1, SRC2)からなる2オペランド方式とされている。このように、命令長を16ビット固定長としたので、コード効率を改善することができる。

【0029】図5(b)は、LSI系(即値演算命令)のフォーマットを示しており、図5(c)は、SHIFT系(シフト演算命令)のフォーマットを示している。また、図5(d)は、SYSCALL系(分岐命令)のフォーマットを示しており、図5(e)は、CFC系(コプロセッサ転送命令)のフォーマットを示している。さらに、図5(f)は、JMP系(分岐命令)のフォーマットを示しており、図5(g)は、LI系(即値演算命令)のフォーマットを示している。

【0030】この他に、ビット処理を行う命令があり、この命令により、例えばビット反転処理等を簡単に行うことができる。従って、コントローラとして用いた場合、効率的に処理を行うことができる。また、上記即値演算命令は、コード効率をさらに改善するのに役立つ。

【0031】図6は、5段のパイプライン動作と、インストラクションバス8、データバス9、およびコプロセッサバス7(伝送手段)の動作を示している。図1に示したように、インストラクションバス8、データバス9、およびコプロセッサバス7は独立のバス構造になっており、入出力も独立になっている。また、各バスは、外部キャッシュ(バッファ)およびコプロセッサ2のレジスタにより結合されている。

【0032】16ビットバスの場合、独立のバス構造にしてもバス幅は大きくならないので、上述したように、コプロセッサバス7をデータバス9と分離して、拡張用コプロセッサの制約をなくすことができる。このように、コプロセッサバス7をデータバス9から分離したことにより、図6に示したように、コプロセッサ2によるロードまたはストア処理と、RISC1による通常のロードまたはストア処理のタイミングを自由に変えることが可能となり、コプロセッサ2をクリティカルパスから開放することができる。また、これにより、コプロセッサ2が自由にバス(コプロセッサバス7)を使用することができるようになる。

【0033】命令デコーダ/パイプライン制御部16は、リセット/クロック制御部19からのクロック信号の立ち下がりのタイミングで、インストラクションバス8をラッチして得られた命令が自分(RISC1)に対

する命令であるのかコプロセッサ2に対する命令であるのかの判定を行い、RISC1に対する命令であると判定した場合、RISC1がその命令を実行し、コプロセッサ2に対応する命令であると判定した場合、その命令に対応する動作をコプロセッサ2のハードウェアによって実現する。このとき、RISC1は何も処理を行わない。

【0034】命令をデコードした後、それぞれのバスの使用可能なタイミングは、図7に示すようになる。このように、コプロセッサ2は、コプロセッサバス7を自由に使用することが可能である。

【0035】また、コプロセッサ2として、第1コプロセッサ(COP0)および第2コプロセッサ(COP1)を拡張することが可能であり、RISC命令によって、汎用レジスタとの間のデータ転送、およびメモリとの間のデータ転送を行うことができる。

【0036】16ビット固定長RISCプロセッサにおいて、5段のパイプラインを停止させることなく、割り込みを効率よく処理させる場合、割り込みテーブルとしては、いわゆる割り込みベクタテーブルより、ジャンプ命令テーブルの方が処理しやすい。図4に示したアドレス空間において、上述したように、ベクタアドレスは4バイトおきに設定され、96バイトの移動可能なジャンプ命令テーブルが配置される。ここには24個の例外を定義することができる。1本の例外を4バイトで定義し、そのうちの2バイト(1ワード)にジャンプ命令が設定され、残りの2バイト(1ワード)はディレイスロットとされる。ここには、ノーオペレーションもしくはその他の命令が設定される。

【0037】いま、図4に示したように、3本の命令例外、7本の内部例外、2本のノンマスカブルインターラップト、および8本の外部割り込みが定義されており、これらを効率よく制御することができる。

【0038】また、外部割り込みを次のように定義することも可能である。すなわち、8本の外部割り込みを、「ラウンドロビン(Round Robin)」モードまたは「フィックス(Fixed)」モードに設定可能である。「ラウンドロビン」モードの場合において、初期値を0に設定したとき、割り込み番号が0, 1, 2, ..., 7, 0, ... の順に所定の時間ずつ巡回的に処理される。また、初期値を1に設定したとき、割り込み番号が1, 2, 3, ..., 7, 0, 1, ... の順に所定の時間ずつ巡回的に処理される。

【0039】さらに、「フィックス」モードの場合においては、初期値を0に設定したとき、0, 1, 2, ..., 7のように順に処理される。初期値を1に設定したとき、1, 2, 3, ..., 7, 0のように順に処理される。

【0040】また、8本の外部割り込みを、ハイ(Hight)とロー(Low)の2つのグループに分け、ハイまたは

ロー側でそれぞれプライオリティを付けることができる。また、このとき、ハイまたはロー側で独立に、「ラウンドロビン」モード、または「フィックス」モードの設定を行うことができる。

【0041】ハイ側の4本の外部割り込みに、「ラウンドロビン」モードを設定した場合においては、初期値を4に設定したとき、割り込み番号が4, 5, 6, 7, 4, 5, ... の順に所定の時間ずつ巡回的に処理が行われる。また、初期値を5に設定したとき、割り込み番号が5, 6, 7, 4, 5, 6, ... の順に所定の時間ずつ巡回的に処理が行われる。

【0042】ハイ側の4本の外部割り込みに、「フィックス」モードを設定した場合においては、初期値を4に設定したとき、割り込み番号が4, 5, 6, 7の順に処理が行われる。また、初期値を5に設定したとき、割り込み番号が5, 6, 7, 4の順に処理が行われる。

【0043】ロー側の4本の外部割り込みに、「ラウンドロビン」モードを設定した場合においては、初期値を0に設定したとき、割り込み番号が0, 1, 2, 3, 0, ... の順に所定の時間ずつ巡回的に処理が行われる。また、初期値を1に設定したとき、割り込み番号が1, 2, 3, 0, 1, ... の順に所定の時間ずつ巡回的に処理が行われる。

【0044】ロー側の4本の外部割り込みに、「フィックス」モードを設定した場合においては、初期値を0に設定したとき、割り込み番号が0, 1, 2, 3の順に処理が行われる。また、初期値を1に設定したとき、割り込み番号が1, 2, 3, 0の順に処理が行われる。

【0045】また、割り込み処理を実行する場合において、図3に示したように、レジスタR31をISP(Interrupt Stack Pointer)として使用することにより、割り込みを多重に管理することが可能になる。すなわち、割り込みが発生すると、PC(Program Counter)をスタック上に退避して外部割り込みを禁止する。これにより、高速な割り込み処理を可能にすることができる。また、RET(I)(Return Interrupt)命令と3重の割り込み許可のスタックを設ける。すなわち、3重の割り込み許可のスタックを設けるようにする。例えば、図3に示したCOP0のレジスタG3(ICR)内に数ビットずつに分割された領域を設け、それらの領域によって3重の割り込み許可のスタックを構成し、割り込みが実行される度にそれらの領域内の値がシフトされるようとする。

【0046】32ビットの固定長命令のRISCでは、一般的に条件分岐命令は1命令で実行される。古典的なCISCでは、比較命令を実行した後、条件分岐命令が実行される。比較命令、および条件分岐命令の実行時間は十分に遅い。これは、比較命令による比較結果に対応してフラグレジスタに確定された値を使用して条件分岐を行っているからである。

【0047】一方、16ビットの固定長RISCにおいては、短い命令長のため、条件分岐命令は1命令では実行できない。そのため、比較命令が実行された後、条件分岐命令が実行される。

【0048】16ビットの固定長命令のRISC1は5段のパイプラインで構成され、フラグレジスタに確定された値を使用すると、比較命令を実行してから条件分岐命令を実行するまでの間、パイプラインディレイが生じてしまう。そこで、比較命令による結果に対応する有効なフラグをバイパス回路12によって構成し、条件分岐命令で即座に使用できるようにしている。これにより、高速な条件分岐が可能になる。

【0049】図8は、5段のパイプライン構成を示している。このように、IF(命令フェッチ)ステージ、RF(レジスタフェッチ)ステージ、ALU(演算)ステージ、MEM(メモリ転送)ステージ、およびWB(レジスタライトバック)ステージより構成される。条件分岐命令の実行時においては、フラグレジスタに値が確定していないパイプライン動作中の最新のアップデートされたフラグが使用される。これは上述したバイパス回路より提供される。最新フラグは、ALU実行時において有効であるか否かが判定され、有効であればその結果が読み出される。また、MEM実行時において、最新フラグが有効であれば、ロードまたはストアされたデータが使用される。このように、各パイプラインステージにおいて、有効フラグであるか否かがチェックされる。

【0050】図9は、16ビット固定長命令のフォーマットを応用したマイクロコードフォーマットの例を示している。一般的にマイクロコードの著作権は認められているが、CPUの命令のバイナリの著作権は認められていない。ところが、RISCの場合、命令体系自体がマイクロコードのフォーマットのようなものであり、これに数ビット(例えば4ビット)を付加して、多目的に使用すると、完全なマイクロコードのフォーマットとができる。ビット15乃至0は、この命令体系そのものである。

【0051】これに、数ビット(ビット-1乃至-4)を付加し、CPU(RISC)1に接続されたハードウェアを多目的に制御することができる。従って、17ビット乃至20ビット長のマイクロコードとして使用することができる。

【0052】複雑なCISC系命令セットの実現には、PLA(Programmable Logic Array)などを用いることによっても可能であるが、一般的にはマイクロコードを用いるのが主流である。マイクロコードのフォーマットは、データパスを効率的に働かせる形式になっている。一方、RISCの命令体系は、固定長命令であるため、マイクロコードのフォーマットとも考えることができ、RISCの命令体系に少し手を加えるだけで、複雑なCISC系命令セットを実現することができる。すなわ

ち、RISCの命令体系は、データパスを効率的に働かせる形式になっている。

【0053】従って、図10に示したように、RISC系命令セットを用いてデータパスを直接制御することが可能である。一方、CISC系命令セットの場合、まず、命令が解析され、次にマイクロコードの先頭のアドレスが計算され、そのアドレスが所定のマイクロコードに与えられ、実行される。なお、マイクロコードフォーマットにおいて、RISC系命令セットに付加される数10ビットは、命令終了ビットなどである。このようにしてデータパスが制御される。

【0054】図11は、上述したようなRISC型マイクロプロセッサ、衛星受信LSI、RAM、およびROMを1チップ化したGPS(Global Positioning System)用の1チップマイコンの構成例を示す図である。1チップマイコン(GPS_LSI)31を構成するブート(BOOT)ROM/ターゲット(Target)ROMデバッガ(Debugger)33は、ブートROMとターゲットROMデバッガより構成され、ブートROMは電源投入時に20実行されるブートプログラムが記憶されている。ターゲットROMデバッガは、デバッガ時に使用されるデバッガプログラムが記憶されている。

【0055】データRAM34は、所定の処理を実行するとき必要とされる各種のデータが記憶される。デュアルポートRAM35は、バス39を介して図示せぬホストコンピュータより送信されてきたデータを記憶したり、ホストに送信するデータを記憶する。インストラクション(Instruction)RAM/ROM36は、所定のアプリケーションプログラムや制御プログラム等を記憶するようになされている。また、SIO(Serial Input/Output)37にPC(Personal Computer)38を接続して、インストラクションRAM/ROM36に記憶されたプログラムのデバッグを行うことができる。

【0056】図12は、本発明の情報処理装置を応用したGPS受信システムの構成例を示している。アンテナ41は図示せぬGPS衛星からの電波を受信し、対応する信号に変換する。帯域通過フィルタ42は、アンテナ41より供給された信号のうち所定の周波数の信号だけを通過させる。アンプ43は、入力された信号を增幅40し、出力するようになされている。

【0057】乗算器44は、アンプ43からの出力信号と後述するC/A(Clear and Acquisition)符号発生器55より供給されたC/A符号を乗算し、出力する。帯域通過フィルタ45は、乗算器44からの出力信号のうち、所定の周波数の信号だけを通過させる。乗算器46は、帯域通過フィルタ45からの出力信号と、後述する倍器53より供給される信号を乗算し、出力する。帯域通過フィルタ47は、乗算器46からの出力信号のうち、所定の周波数の信号だけを通過させる。乗算器48は、帯域通過フィルタ47からの出力信号と、後

述するてい倍器54より供給される信号を乗算し、出力する。帯域通過フィルタ49は、乗算器48より供給された信号のうち、所定の周波数の信号だけを通過させる。

【0058】同期追跡回路50は、PLL等で構成され、同期検出を行い、同期検出信号や再生搬送波を出力するようになされている。雑音検出フィルタ51は、入力信号に含まれる雑音を検出し、除去するようになされている。同期捕捉回路52は、入力された同期検出信号や再生搬送波に基づいて、クロック信号を発生し、出力する。てい倍器53、54は、それぞれ入力された信号の高調波を取り出し、増幅して出力する。C/A符号発生器55は、同期捕捉回路52より供給されたクロック信号に同期して、C/A符号、すなわちPN(Pseudo-Noise)符号を発生し、出力するようになされている。

【0059】次に、その動作を説明する。GPS衛星より送信されたGPS電波は、アンテナ41により受信され、所定の信号に変換され、アンプ43により増幅された後、乗算器44に供給される。GPS衛星は、搬送波にC/A符号を乗算してスペクトラム拡散している。従って、受信側においては、GPS衛星が使用したものと同一のC/A符号を受信信号に乗算し、狭帯域に戻す処理が行われる。すなわち、乗算器44においては、アンプ43より供給された信号と、C/A符号発生器55より供給されたC/A符号が乗算され、帯域通過フィルタ45に供給される。

【0060】帯域通過フィルタ45においては、そこに供給された信号のうち、所定の周波数の信号だけが通過され、乗算器46に供給される。乗算器46においては、帯域通過フィルタ45より供給された信号と、てい倍器53より供給された信号とが乗算され、帯域通過フィルタ47に供給される。帯域通過フィルタ47においては、そこに供給された信号のうち、所定の周波数の信号だけが通過され、乗算器48に供給される。

【0061】乗算器48においては、帯域通過フィルタ47より供給された信号と、てい倍器54より供給された信号とが乗算され、帯域通過フィルタ49に供給される。帯域通過フィルタ49においては、そこに供給された信号のうち、所定の周波数の信号だけが通過され、同期追跡回路50に供給される。同期追跡回路50に入力された信号は、まず、雑音検出フィルタ51に供給され、信号に含まれる雑音が除去される。雑音が除去された信号は再び同期追跡回路50に供給され、次に、入力信号の同期追跡が行われ、同期検出信号や再生搬送波が出力される。

【0062】同期追跡回路50より出力された同期検出信号および再生搬送波は、同期捕捉回路52に供給される。同期捕捉回路52においては、同期追跡回路50より供給された同期検出信号や再生搬送波に基づいて、所定の周期のクロック信号が発生され、C/A符号発生器

55に供給される。C/A符号発生器55においては、そこに供給されたクロック信号に同期してC/A符号が発生され、乗算器44に供給される。このとき、C/A符号発生器55より出力されるC/A符号の位相は、少しずつずらされ、受信信号の位相と一致するように調整される。

【0063】このようにして、受信信号はベースバンド信号に変換され、復調データを得る。得られた復調データは、同期追跡回路50より図示せぬCPUに供給される

10

【0064】上述したような動作をするGPS受信システムを、1つのチップで構成することも可能である。そして、それを上述した図11の衛星受信LSI32として使用するようにすることができる。

【0065】その場合、衛星受信LSI32によって受信されたGPS衛星からの信号は、RISC1に供給される。RISC1は、衛星受信LSI32より供給された受信信号に基づいて、所定の演算処理を高速に実行し、現在の位置を計測する。次に、現在の位置に対応する地図情報を図示せぬCD-ROMなどより読み出し、画像データに変換した後、対応する地図を図示せぬCRTに表示する。また、進行方向やユーザの操作に対応して、画像データを2次元的または3次元的に座標変換して、表示することもできる。また、交差点に接近したときなどに所定の音声信号を合成し、所定のタイミングで出力するようにすることも可能である。

【0066】このように、RISC1と衛星受信LSI32を1チップ化することにより、ナビゲーションシステムを1チップで構築することが可能である。また、1チップ化することにより、装置のコストを削減することができ、消費電力を抑えることが可能となる。

【0067】なお、上記実施例においては、パイプライン数を5段としたが、これに限定されるものではない。

【0068】また、上記実施例においては、命令長を16ビット固定長としたが、他のビット数に固定することもできる。

【0069】また、上記実施例において、RISC1の命令をマイクロコードとして使用する場合に、その命令に新たに付加するビット数を4ビットとしたが、他の任意のビット数とすることも可能である。

【0070】また、上記実施例においては、GPS受信システムに使用可能なRISCプロセッサを用いた1チップマイコンの構成例を示したが、GPS受信システムに限定されるものではなく、例えば、ゲーム機や携帯型情報通信機器、あるいはカラオケ装置などのマルチメディア機器等、その他の機器に適用することも可能である。

【0071】

【発明の効果】請求項1に記載のRISC型マイクロプロセッサによれば、デコード手段により、所定の固定長

の命令がデコードされるようにしたので、命令のデコード処理を高速化することができる。従って、演算処理を高速化することが可能となる。

【0072】請求項6に記載のRISC型マイクロプロセッサによれば、モード指定手段により、複数の割り込みがあったとき、割り込みに対応する処理をそれぞれ所定の時間ずつ巡回的に実行する第1のモードと、割り込みに対応する処理を逐次実行する第2のモードのいずれかのモードが指定され、このモードに従って、割り込み制御手段により割り込み処理が制御されるようにしたので、効率よく割り込み処理を実行させることができる。従って、コントローラとして使用することが可能となる。

【0073】請求項8に記載のRISC型マイクロプロセッサによれば、RISC型マイクロプロセッサのデータバスとは独立して、コプロセッサが専用に使用し、データの伝送を行う伝送手段が設けられるようにしたので、コプロセッサは、RISC型マイクロプロセッサとは独立してデータ伝送を行うことができる。従って、コプロセッサをクリティカルパスから開放することが可能となる。

【0074】請求項9に記載の情報処理装置によれば、GPS衛星からの電波を受信する受信装置と、受信装置により受信された電波に対応する信号を処理するRISC型マイクロプロセッサが1チップ化されているようにしたので、装置を小型化することができ、消費電力を低減させることができるとなる。

【図面の簡単な説明】

【図1】本発明のRISC型マイクロプロセッサを応用した1チップマイコンの一実施例の構成を示すブロック図である。

【図2】図1のRISC1の内部構成を示す図である。

【図3】図1の1チップマイコンのレジスタ構成を示す図である。

【図4】RISC1のアドレス空間を示す図である。

【図5】RISC1において用いられる命令フォーマットを示す図である。

【図6】命令デコーダ／パイプライン制御部16のパイプライン動作を示した図である。

【図7】コプロセッサバス7の動作を示した図である。

【図8】パイプライン構成を示す図である。

【図9】マイクロコードのフォーマットを示す図である。

* 【図10】CISC系命令とRISC系命令によるデータバスの制御方法の違いを説明するための図である。

【図11】GPS-LSIの構成例を示すブロック図である。

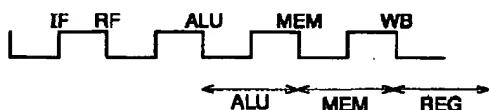
【図12】GPS受信システムの構成例を示すブロック図である。

【符号の説明】

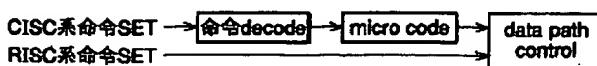
- | | |
|----------------|-------------------------------|
| 1 | RISC |
| 2 | コプロセッサ (Co_pro) |
| 10 | IPL_ROM |
| 4 | コードバッファ (Code buffer) |
| 5 | データバッファ (Data buffer) |
| 6 | メモリコントローラ (Memory controller) |
| 7 | コプロセッサバス (Co_pro bus) |
| 8 | インストラクションバス (Instruction bus) |
| 9 | データバス (Data bus) |
| 11 | 汎用レジスタ |
| 12 | バイパス回路 |
| 13 | ALU |
| 20 | 14 高速乗算器／高速除算器 |
| 15 | アドレス計算部 |
| 16 | 命令デコーダ／パイプライン制御部 |
| 17 | 専用コントロールレジスタ |
| 18 | 割り込みコントローラ |
| 19 | リセット・クロック制御部 |
| 31 | GPS_LSI |
| 32 | 衛星受信LSI |
| 33 | ブートROM／ターゲットROMデバッガ |
| 34 | データRAM |
| 30 | 35 デュアルポートRAM |
| 36 | インストラクションRAM／ROM |
| 37 | SIO |
| 38 | PC |
| 41 | アンテナ |
| 42, 45, 47, 49 | 帯域通過フィルタ |
| 43 | アンプ |
| 44, 46, 48 | 乗算器 |
| 50 | 同期追跡回路 |
| 51 | 雑音検出フィルタ |
| 40 | 52 同期捕捉回路 |
| 53, 54 | てい倍器 |
| 55 | C/A符号発生器 |

*

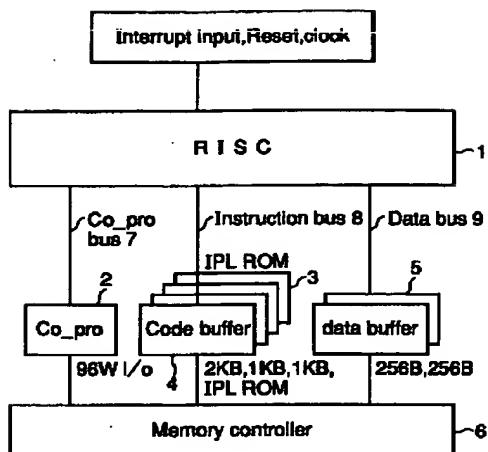
【図8】



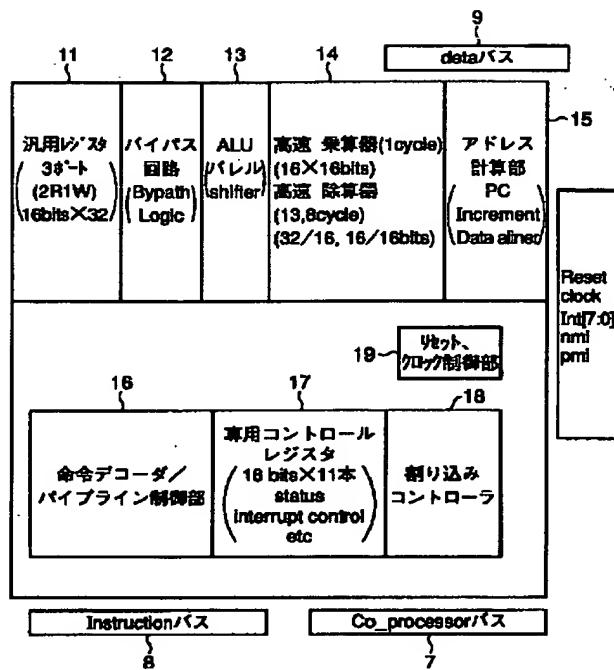
【図10】



【図1】



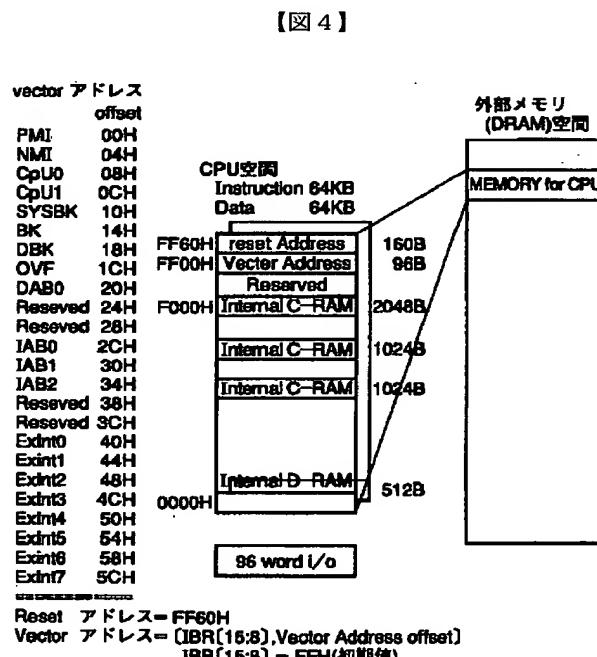
【図2】



【図3】

CPU register	COP0 register	COP0 register	COP1 register	COP1 register
R0	G0 SR	C0	G0	C0
R1 ACC	G1 MCR	C1	G1	C1
R2	G2 IBR	C2	G2	C2
R3	G3 ICR	C3	G3	C3
R4	G4 IMR0	C4	G4	C4
R5	G5 IMR1	C5	G5	C5
R6	G6 JBR	C6	G6	C6
R7	G7	C7	G7	C7
R8	G8 DAB0	C8	G8	C8
R9	G9	C9	G9	C9
R10	G10	C10	G10	C10
R11	G11 TAB1	C11	G11	C11
R12	G12 TAB2	C12	G12	C12
R13	G13	C13	G13	C13
R14	G14	C14	G14	C14
R15	G15	C15	G15	C15
R16	G16	C16	G16	C16
R17	G17	C17	G17	C17
R18	G18	C18	G18	C18
R19	G19	C19	G19	C19
R20	G20 XDDD	C20	G20	C20
R21	G21	C21	G21	C21
R22	G22	C22	G22	C22
R23	G23	C23	G23	C23
R24	G24	C24	G24	C24
R25	G25	C25	G25	C25
R26	G26	C26	G26	C26
R27	G27	C27	G27	C27
R28	G28	C28	G28	C28
R29	G29	C29	G29	C29
R30 SP	G30	C30	G30	C30
R31 ISP	G31	C31	G31	C31

H1 L0

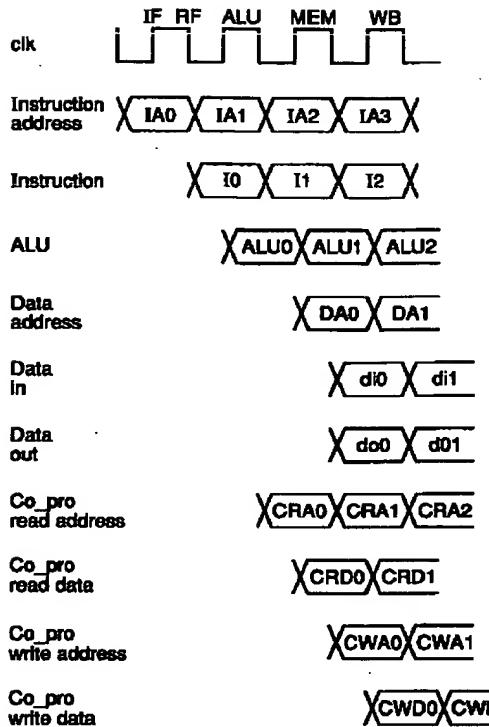


Reset アドレス = FF60H
 Vector アドレス = [IBR(15:8), Vector Address offset]
 IBR(15:8) = FFH(初期値)

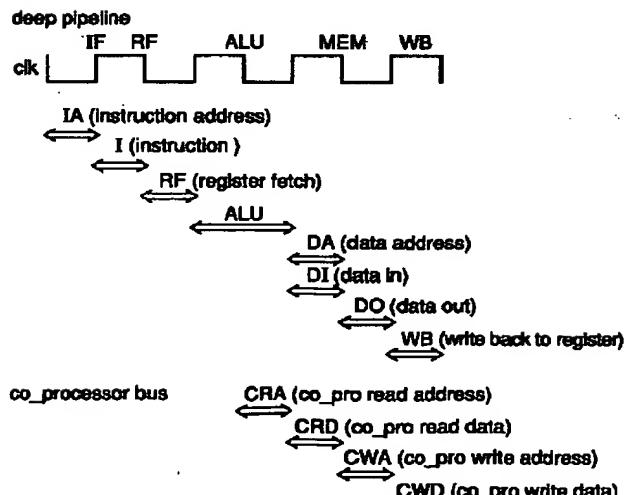
【図5】

(a) MOV系	<table border="1"> <tr><td>15</td><td>10 9</td><td>5 4</td><td>0</td></tr> <tr><td>OP_CODE</td><td>SRC1/DEST1</td><td>SRC2</td><td></td></tr> </table>	15	10 9	5 4	0	OP_CODE	SRC1/DEST1	SRC2	
15	10 9	5 4	0						
OP_CODE	SRC1/DEST1	SRC2							
(b) LSI系	<table border="1"> <tr><td>15</td><td>10 9</td><td>5 4</td><td>0</td></tr> <tr><td>OP_CODE</td><td>SRC1/DEST1</td><td>即値</td><td></td></tr> </table>	15	10 9	5 4	0	OP_CODE	SRC1/DEST1	即値	
15	10 9	5 4	0						
OP_CODE	SRC1/DEST1	即値							
(c) SHIFT系	<table border="1"> <tr><td>15</td><td>10 9</td><td>5 4 3</td><td>0</td></tr> <tr><td>OP_CODE</td><td>SRC1/DEST1</td><td>即値</td><td></td></tr> </table>	15	10 9	5 4 3	0	OP_CODE	SRC1/DEST1	即値	
15	10 9	5 4 3	0						
OP_CODE	SRC1/DEST1	即値							
(d) SYSCALL系	<table border="1"> <tr><td>15</td><td>10 9</td><td>5 4</td><td>0</td></tr> <tr><td>OP_CODE</td><td>reserved</td><td>OP_CODE</td><td></td></tr> </table>	15	10 9	5 4	0	OP_CODE	reserved	OP_CODE	
15	10 9	5 4	0						
OP_CODE	reserved	OP_CODE							
(e) CFC系	<table border="1"> <tr><td>15</td><td>10 9</td><td>5 4</td><td>0</td></tr> <tr><td>OP_CODE</td><td>reserved</td><td>OP_CODE</td><td></td></tr> </table>	15	10 9	5 4	0	OP_CODE	reserved	OP_CODE	
15	10 9	5 4	0						
OP_CODE	reserved	OP_CODE							
(f) JMP系	<table border="1"> <tr><td>15</td><td>10 9</td><td>0</td></tr> <tr><td>OP_CODE</td><td>OFFSET</td><td></td></tr> </table>	15	10 9	0	OP_CODE	OFFSET			
15	10 9	0							
OP_CODE	OFFSET								
(g) I/O系	<table border="1"> <tr><td>15</td><td>10 9 8 7</td><td>0</td></tr> <tr><td>OP_CODE</td><td></td><td>即値</td></tr> </table>	15	10 9 8 7	0	OP_CODE		即値		
15	10 9 8 7	0							
OP_CODE		即値							

【図6】



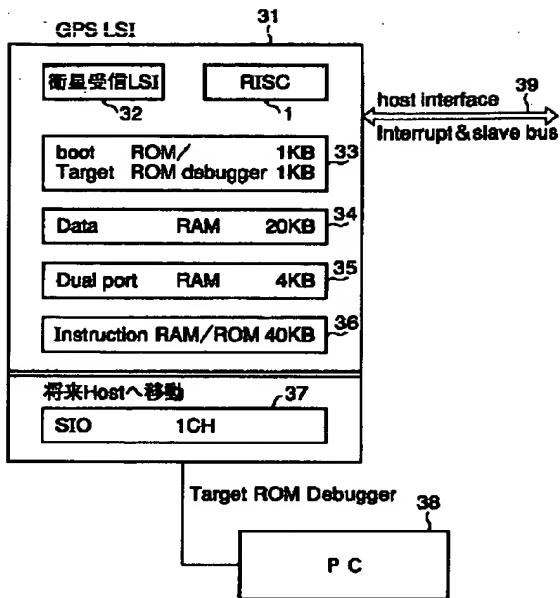
【図7】



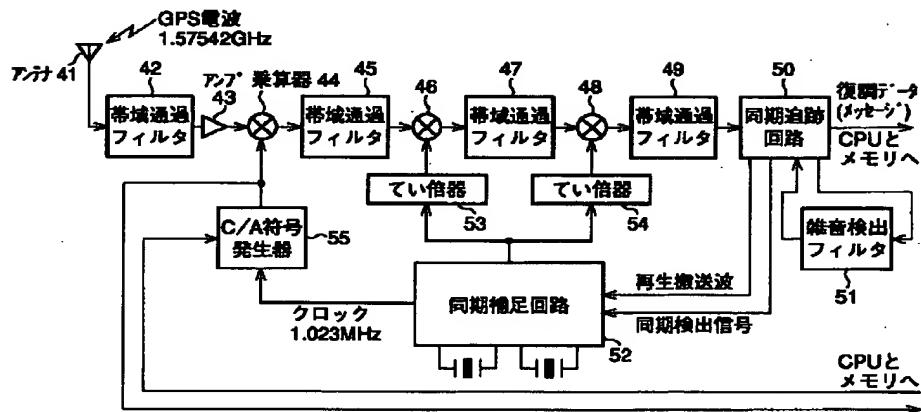
【図9】

(a) MOV系	<table border="1"> <tr><td>15</td><td>10 9</td><td>5 4</td><td>0 -1 -4</td></tr> <tr><td>OP_CODE</td><td>SRC1/DEST1</td><td>SRC2</td><td></td></tr> </table>	15	10 9	5 4	0 -1 -4	OP_CODE	SRC1/DEST1	SRC2	
15	10 9	5 4	0 -1 -4						
OP_CODE	SRC1/DEST1	SRC2							
(b) LSI系	<table border="1"> <tr><td>15</td><td>10 9</td><td>5 4</td><td>0 -1 -4</td></tr> <tr><td>OP_CODE</td><td>SRC1/DEST1</td><td>即値</td><td></td></tr> </table>	15	10 9	5 4	0 -1 -4	OP_CODE	SRC1/DEST1	即値	
15	10 9	5 4	0 -1 -4						
OP_CODE	SRC1/DEST1	即値							
(c) SHIFT系	<table border="1"> <tr><td>15</td><td>10 9</td><td>5 4 3</td><td>0 -1 -4</td></tr> <tr><td>OP_CODE</td><td>SRC1/DEST1</td><td>即値</td><td></td></tr> </table>	15	10 9	5 4 3	0 -1 -4	OP_CODE	SRC1/DEST1	即値	
15	10 9	5 4 3	0 -1 -4						
OP_CODE	SRC1/DEST1	即値							
(d) SYSCALL系	<table border="1"> <tr><td>15</td><td>10 9</td><td>5 4</td><td>0 -1 -4</td></tr> <tr><td>OP_CODE</td><td>reserved</td><td>OP_CODE</td><td></td></tr> </table>	15	10 9	5 4	0 -1 -4	OP_CODE	reserved	OP_CODE	
15	10 9	5 4	0 -1 -4						
OP_CODE	reserved	OP_CODE							
(e) CFC系	<table border="1"> <tr><td>15</td><td>10 9</td><td>5 4</td><td>0 -1 -4</td></tr> <tr><td>OP_CODE</td><td>reserved</td><td>OP_CODE</td><td></td></tr> </table>	15	10 9	5 4	0 -1 -4	OP_CODE	reserved	OP_CODE	
15	10 9	5 4	0 -1 -4						
OP_CODE	reserved	OP_CODE							
(f) JMP系	<table border="1"> <tr><td>15</td><td>10 9</td><td>0 -1 -4</td></tr> <tr><td>OP_CODE</td><td>OFFSET</td><td></td></tr> </table>	15	10 9	0 -1 -4	OP_CODE	OFFSET			
15	10 9	0 -1 -4							
OP_CODE	OFFSET								
(g) I/O系	<table border="1"> <tr><td>15</td><td>10 9 8 7</td><td>0 -1 -4</td></tr> <tr><td>OP_CODE</td><td></td><td>即値</td></tr> </table>	15	10 9 8 7	0 -1 -4	OP_CODE		即値		
15	10 9 8 7	0 -1 -4							
OP_CODE		即値							

【図1-1】



【図1-2】



**This Page is Inserted by IFW Indexing and Scanning
Operations and is not part of the Official Record**

BEST AVAILABLE IMAGES

Defective images within this document are accurate representations of the original documents submitted by the applicant.

Defects in the images include but are not limited to the items checked:

- BLACK BORDERS**
- IMAGE CUT OFF AT TOP, BOTTOM OR SIDES**
- FADED TEXT OR DRAWING**
- BLURRED OR ILLEGIBLE TEXT OR DRAWING**
- SKEWED/SLANTED IMAGES**
- COLOR OR BLACK AND WHITE PHOTOGRAPHS**
- GRAY SCALE DOCUMENTS**
- LINES OR MARKS ON ORIGINAL DOCUMENT**
- REFERENCE(S) OR EXHIBIT(S) SUBMITTED ARE POOR QUALITY**
- OTHER:** _____

IMAGES ARE BEST AVAILABLE COPY.

As rescanning these documents will not correct the image problems checked, please do not report these problems to the IFW Image Problem Mailbox.